7th Annual Conference on Systems Engineering Research 2009 (CSER 2009)

Determining the Lines of System Maturity, System Readiness and Capability Readiness in the System Development Lifecycle

Mr. Abideen Tetlay[1] and Prof. Philip John[2]

[1] Cranfield University (UK), abideen.tetlay@cranfield.ac.uk
[2] Cranfield University (UK), p.john@cranfield.ac.uk

Abstract

Many systems suffer from major problems with implementation, integration, performance and other lifecycle problems and unexpected and unacceptable behaviour often arises when they are introduced into the real world, yet the systems were believed to be "ready" for use. Often, certain aspects about the real world context are not always appreciated until surprise problems arise when the system is introduced and used. This could be thought of as a failure in "systems understanding". We are not good enough at understanding system implementation and integration issues (in their widest sense) and the associated Risk issues in assessing a System's Maturity and Readiness within a development programme and overall lifecycle. This challenge is increasing interest and emphasis on System Maturity assessments and in the use of System Readiness Levels and on achieving the overall Capability of a system in its intended operational environment.

In this paper, we investigate the issues by addressing the following key questions: Why are "Maturity" and "Readiness" important in Systems Engineering (SE)? Is there a sufficiently clear distinction between 'System Maturity' (SM) and 'System Readiness' (SR)? What do we mean by SM and SR? How is the term 'Capability' currently being used in SE? Why do we need to assess the Capability of the system? What do we mean by 'Capability Readiness' (CR)? We define SM, SR and CR and map these to the System Development and overall Lifecycle and then provide recommendations for further research.

Key words – System Maturity; System Readiness; Capability Readiness; System Development Lifecycle.

1 Introduction

The Twenty-first Century has been called "The Systems Century". In an increasingly, highly-integrated technological world the nature of Systems Design and its conduct within engineering projects is also increasingly complex. Complex systems products cannot be designed in isolation and decisions must be made in the face of increasing complexity, uncertainty and rapid change.

The crucial objective of a lifecycle is to achieve a system that is "successful" in use in the "real world" and decisions made in the development of a system must be made with this aim and hence, they must understand what "success" would be (to different stakeholder perspectives) and what real world context and environment will the system be exposed to / interoperate with / be affected by and how would a potential system concept behave in such a context. These are issues that are often causes of problems in real world projects.

During the development of a system, assessing the "maturity" of the system definition towards a successful outcome is important, as is the assessment of the "readiness" of a system to undertake roles within the real world.

2 Methodology

We address the issues associated with "maturity" and "readiness" by providing answers to the following key questions:

1) Why are "Maturity" and "Readiness" important in Systems Engineering (SE)?

2) Is there a sufficiently clear distinction between 'System Maturity' (SM) and 'System Readiness' (SR)?

3) What do we mean by SM and SR?

4) How is the term 'Capability' currently being used in SE?

5) Why do we need to assess the Capability of the system?

6) What do we mean by 'Capability Readiness' (CR)?

3 Why are "Maturity" and "Readiness" important in Systems Engineering?

The issue of "Maturity" and "Readiness" and its importance within the systems engineering decision making process is clearly evident with the increasing interest in recent years in the ability to judge "metrics" such as "Technology Readiness Levels" (TRLs) [01], [02], "System Readiness Levels" (SRLs) [03], [04], "Integration Readiness Levels" (IRLs) [05] and "Manufacturing Readiness Levels" (MRLs) [06], [07].

However, the recent work [08], [09], [10], [11] has also exposed the difficulty of achieving meaningful, clear and quantifiable metrics. One aspect of this difficulty arises because the notions of "maturity" and "readiness" are completely meaningful only within a contextual setting: "mature enough for what?" and "ready for what?" Therefore, we need to be able to judge and express a system's Maturity to assess when we have achieved a defined and implemented system. Questions arise about the form with which a meaningful, useful metric for Maturity would take. What issues should be considered in forming the Maturity judgement and how it should be expressed? A meaningful view of the Maturity of "a system" is key to the determination of Risks associated with its development and operation. This notion applies at all system levels: for the integration of a system into a wider environment and other systems, for the design of a system itself and for any particular part of that system.

We want to reach a situation where we are more confident that we shall not experience the unexpected and unacceptable problems we have often seen in the lifecycle. This has led to an increasing emphasis in assessing "Maturity" and "Readiness". We need to judge Readiness to assess when a system is fit for purpose for a particular context.

4 Is there a sufficiently clear distinction between 'System Maturity' (SM) and 'System Readiness' (SR)?

Current definitions of **Readiness** seem to assess SM in order to determine SR. If you look at the formal definitions below for TRLs, SRLs and IRLs for example, you will note that they all refer to the key term of ***"maturity"*** in their definitions:

TRLs: *"TRLs are a Technology Management tool that provides an indication of the technical **maturity** of a project by identifying risk associated with technology and system integration. They are a graduated scale that uses specific criteria to define the **maturity** of technology. TRLs were developed by NASA in the 1980s and in 2001 Mckinsey recommended that the MOD adopt their use [12].*"

TRLs: *"A systematic metric/measurement system that supports assessment of the **maturity** of a particular technology and the consistent comparison of **maturity** between different types of technologies (J. C. Mankins, 2002) [13].*"

SRLs: *"System Readiness Levels (SRLs) have been developed as a project management tool to capture evidence, and assess and communicate **System Maturity** in a consistent manner to stakeholders. SRLs define a set of **nine maturity steps** from Concept to in-service across a set of systems engineering disciplines. Each of the SRL steps align to key outputs from systems disciplines such as Training, Safety and Environmental, or Reliability and Maintainability. SRLs track a project's progress against the systems engineering 'V' diagram [14].*"

IRLs: *"A systematic measurement of the interfacing of compatible interactions for various technologies and the consistent comparison of the **maturity** between integration points [13].*"

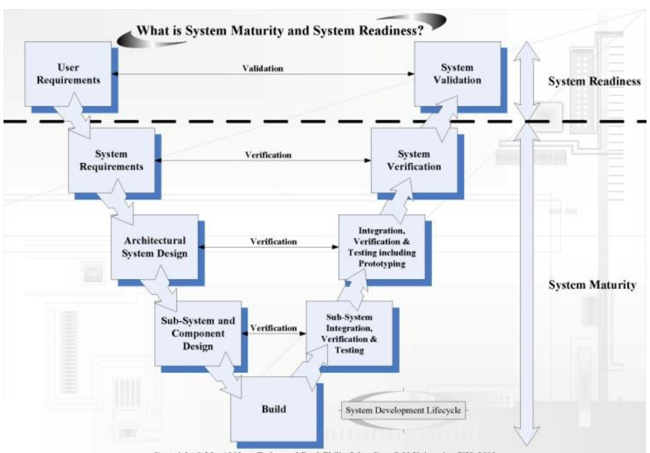
Based on this evidence, you could infer that the notion of ***"maturity"*** is encapsulated within the notion of ***"readiness"***. This explains why in texts they are not considered to be two distinct and separate entities and are not used in isolation, but interchangeably. Maturity is therefore regarded as a part of Readiness (in software/UML terms, an aggregation type of relationship) and you cannot talk about *readiness* without discussing or mentioning *maturity* and vice versa; they are not mutually exclusive. It could be argued that the existing Readiness Levels actually provide a "Maturity" metric as opposed to a "Readiness" metric.

According to [15], "**"Readiness"** values tend to be *soft metrics* that are: relatively easy to derive, but require a complementing rationale that explains the assessment, human-intensive, subjective, contain inherent variations or ambiguity that is averaged away." According to [16], "the common methods to determine any Readiness Levels (xRLs) is to use an individual expert: an expert assesses its state of **maturity**; Group: the **maturity** is determined through a discussion among the stakeholders; and Assessment Tool: the use of guidance documentation or a software tool that directs the **maturity assessment**."

5 What do we mean by SM and SR?

We would argue that the notions of **"maturity'** (System Maturity) and **"readiness"** (System Readiness) should be treated as two clear and distinct entities both of which are actually addressing two completely different questions within the scope and context of the System Development and overall Lifecycle. To illustrate this argument, we have mapped SM and SR against the System Development and overall Lifecycle as depicted in Figure 1. We also provide a definition for SM and SR:

Figure 1 - What is System Maturity and System Readiness?



We would define **'System Maturity'** as follows:

"System Maturity (SM) is the ***verification*** *within an* ***iterative*** *process of the system development lifecycle and occurs before System Readiness, i.e. the system must first be fully 'mature' before it can be 'ready' for use. The process starts from System Requirements and finishes at System Verification. System Maturity asks the question: Do we have a complete, well defined design that has been implemented and verified (i.e. we have decided what we want to implement and we have achieved this; the designed product now physically exists)?*

Three phases or states of 'System Maturity' could be envisaged:

- *System is Immature (SI) - have not started yet and have not produced anything (for example, at the System Requirements stage of the System Development Lifecycle).*

- *System Maturity is in Progress (SMP) - working your way through the System Development Lifecycle (for example, the design, development and testing part of the System Development Lifecycle) in order to decide and define the system design and bring it into existence.*

- *System Maturity has been Achieved (SMA) - the design, development and testing of the system is now complete, fully 'mature' and tested. To achieve System Maturity the System must be verified against the System Requirements, i.e. you have achieved SM by building the system right. (Tetlay and John, 2009)".*

Verification confirms that the system element meets the design-to or build-to specifications. It answers the question "Did you build it right? [17]".

We would define **'System Readiness'** as follows:

"System Readiness (SR) is the ***validation*** *and* ***Boolean*** *(either the system is 'ready' for use or not) aspect of the system development and overall lifecycle and occurs after System Maturity, i.e. the system must first be fully 'mature' before it can be made 'ready' for use. The process starts from User Requirements and finishes at System Validation. System Readiness determines whether or not the system is now 'ready' for use in its intended* ***operational environment****. Therefore, System Readiness is* ***context dependent****. To achieve System Readiness the System must be validated against the User Requirements, i.e. you will achieve SR by building the right system for a given context (Tetlay and John, 2009)".*

Validation answers the question of "Did you build the right thing? [17]". Note that this question is implicitly context dependent, i.e. "right" for what?

6 How is the term 'Capability' currently being used in SE?

First, here are some definitions for the term 'Capability':

*"A measure of the system's **ability** to achieve the **mission objectives**, given that the system is dependable and suitable. Examples of 'capability measures' are: accuracy; range; payload; lethality; information rates; number of engagements; destructiveness; design constraints; and/or technical exit criteria. Capability is a systems engineering metric [18]."*

*"Capability is the enduring **ability** to generate a desired **operational outcome** or effect, and is relative to the threat, **physical environment** and the contributions of coalition partners. Capability is not a particular system or equipment [19]".*

We prefer the second definition from [19] because it emphasis the fact that 'Capability' is associated with the System's ability to produce an **operational outcome**, but also taking into consideration other factors, such as the **environment** which may affect the 'Capability' of the overall system operating in the real world. Also, note that both definitions emphasise that 'Capability' has a strong association with the **ability** of the system to achieve an **operational outcome** which we concur with.

Next, we will investigate the use of ***Capability Models*** to determine how 'Capability' is often used. According to [20], "several versions of a systems engineering assessment model were developed, starting as early as 1993 with the development of the Capability Assessment Model for Systems Engineering (Widmann, 1993) and the Systems Engineering Maturity Model (Mar, 1993). These early models grew out of the work of the INCOSE Capability Assessment Working Group (CAWG) (Widmann, 1993; Mar, 1993). Soon, there was a growing effort in the systems engineering community to develop a systems engineering maturity model like the one developed for software engineering by the Software Engineering Institute (SEI) at Carnegie Mellon University (Brill and Brammer, 1993; Kuhn and Garcia, 1994). By 1995, the CAWG had released the Systems Engineering Capability Assessment Model (SECAM), which was being used for self-assessments in companies such as Computer Sciences Corporation (Mackay, 1993). Concurrently, the Industry Collaboration Group (later EPIC) had gone forward with a modified Software CMM model for systems engineering, known as the SECMM (Kuhn, 1985). By 1996, companies such as Texas Instruments had employed the SECMM to perform assessments of their own (Kuhn, 1996). Allied Signal had developed their own systems engineering assessment models for consideration in the systems engineering community (Booth and Oran, 1996). The systems engineering community recognised it would be best if different systems engineering capability assessment models were combined into one accepted industry standard model (Widmann et al., 1996). To a degree, this was accomplished in 1997 by the release of a draft combined assessment model known as EIA/IS-731-1 Systems Engineering Capability Model (SECM) (Widmann and Mindlin, 1998). This model served as a foundation for the development of another merged assessment standard based on the work performed by the SEI in developing the Software Capability Maturity Model (SW-CMM) (Ahern, 1999; Schoening and Clouse, 1999). The Capability Maturity Model Integration (CMMI) project, sponsored by the US Secretary of Defense for Acquisition and Technology, merged assessment models for software engineering, systems engineering, project management, and integrated product and process development (IPPD) in both a staged and continuous model representation (Ferguson, 1999). The latest version of the CMMI model is Version 1.02, dated November 2000 (SEI, 2000)".

In précis, historically, systems (and software) engineers have used 'Capability' in the form of process improvement of systems (or software) engineering processes and not within the context of the operational outcome of the system, i.e. the Capability of the system operating in its environment for a given real world context. It is important to note that we are only interested in using the term 'Capability' in this latter context.

7 Why do we need to assess the Capability of the system?

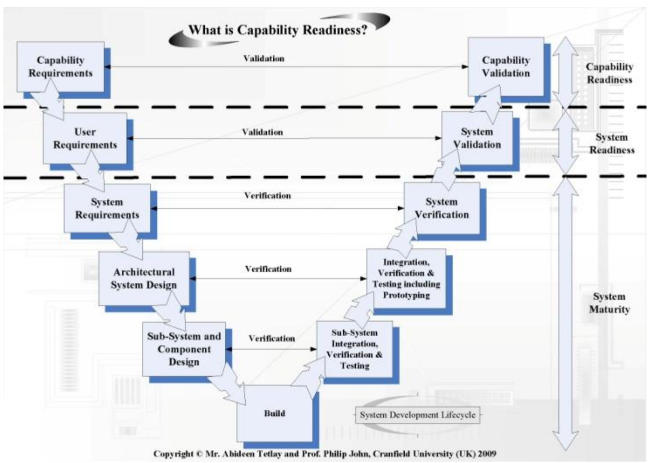
The concept of **Readiness** may be further expanded and related to **Capability**. We need to assess the Capability of the system because if we used for example, the US Defense Acquisition Life Cycle Framework, any Readiness Levels (xRLs) only make an assessment from Pre-Concept through to Production and Deployment. **There is no 'formal' assessment of the 'operational and support capability' of the system** [07]. However, Manufacturing Readiness Levels (MRLs) do attempt to address this, but from a manufacturing perspective.

TRLs or IRLs for example, only assess the 'Technology' or 'Integration' aspects of the system, rather than other key aspects which are equally important and are also likely to affect the 'Capability' of the system operating in the real world. Other aspects could include the Defence Lines of Development (DLoD) – TEPIDOIL [21] for example. According to [11], "**TRL is not an end state to determining a system's readiness based on: TRL is only a measure of an individual technology and not systems readiness. There is no proven, tested, systematic index of systems readiness**".

8 What do we mean by 'Capability Readiness' (CR)?

In the illustration in Figure 2, we have mapped 'Capability Requirements' at the very beginning of the V-Model and before User Requirements which has traditionally been the starting point of the model. The premise for this is to ensure that we capture the *full* "complete" requirements starting from and including the 'Capability Requirements' which we need to build and factor into the System Development and overall Lifecycle.

Figure 2 - What do we mean by CR?



We would define '**Capability Readiness'** as follows:

"Capability Readiness (CR) determines whether or not the system has the ***ability*** *and the* ***capacity*** *to completely fulfil the* ***operational capability*** *of the system for a given context in its intended* ***operational environment*** *within the* ***scope*** *of the* ***Capability Requirements*** *and its aims and objectives. Once we know that the system has achieved System Readiness then we can raise the Capability Readiness question. Like System Readiness, Capability Readiness is looking at the* ***validation*** *of the system and is also* ***context dependent****. The process starts at Capability Requirements and finishes at Capability Validation. To achieve Capability Readiness the system must be validated against the Capability Requirements, i.e. you will achieve CR if you can "demonstrate" (using the Defence Lines of Development (DLoD) – TEPIDOIL for example) that the system does have the ability and the capacity to completely fulfil the operational capability of the system for its intended operational environment as prescribed by the Capability Requirements (Tetlay and John, 2009)".*

9 Conclusions

"Maturity" and "Readiness" are important in Systems Engineering because we need to be able to judge and express a system's Maturity to assess when we have achieved a defined and implemented system. We need to judge Readiness to assess when a system is fit for purpose for a particular context. A meaningful view of the Maturity of a "system" is key to the determination of Risks associated with its development and operation and the progress that has been made in dealing with them and achieving an end system product.

However, the notion of *"maturity"* appears to be encapsulated within the notion of *"readiness"* and they are often used interchangeably. This is likely to lead to the confusion in the understanding of these terms and in the use of existing Readiness Levels. Existing Readiness Levels do not provide an 'end state' to determining a system's readiness and there is no proven, tested, systematic index of real systems readiness.

We propose that it is both useful and necessary to treat *"maturity"* as distinct from *"readiness"*. *'System Maturity'* is to do with the *verification* within an *iterative* process of the System Development Lifecycle. System Maturity is focusing on the design maturity of a system product and is only verified against the System Requirements if it is successfully implemented as intended by the design. Whereas, *'System Readiness'* is to do with the *validation* and *Boolean* (either the system is 'ready' for use or not) aspect of the System Development and overall Lifecycle as well as being *context dependent*. System Readiness validates whether or not the system can satisfy the User Requirements for a given context.

'Capability Readiness' determines whether or not the system has the *ability* and the *capacity* to completely fulfil the *operational capability* of the system for a given context in its intended *operational environment* within the *scope* of the *Capability Requirements* and its aims and objectives. Like System Readiness, Capability Readiness is looking at the *validation* of the system and is also *context dependent*. Capability Readiness validates (against the Defence Lines of Development (DLoD) – TEPIDOIL for example) whether or not all aspects of a system are available for a given context.

Historically, systems (and software) engineers have used 'Capability' in the form of process improvement of systems (or software) engineering processes and not within the context of the operational outcome of the system, i.e. the Capability of the system operating in its environment for a given real world context. It is important to note that we are only interested in using the term ‘Capability’ in this context.

We need to assess the Capability of the system because there is no ‘formal’ assessment and measurement for the Capability of the system for a given context in its intended operational environment. There is no proven, tested, systematic index of ‘Capability Readiness’. We need novel approaches for evaluating the progress of design decisions towards a successful “Capability” operating in the real world.

10 Recommendations

We provide the following recommendations for further research:

- Create a new set of ‘Capability Readiness Levels’ (CRLs) to assess and measure a system for a given context in its intended operational environment, i.e. to determine the degree of ‘Capability Readiness’ (CR).

- Establish a clear, useful Framework for assessing and measuring ‘Capability Readiness’ (CR), including the development of a rigorous “Metric”, a process for its use within a development programme and overall lifecycle and how it applies at different System Levels, from individual Subsystem to Networked Systems of Systems (SoS).

- Provide a link between the CRLs and the CR Framework in the form of a Decision Tree.

- Apply and test the CRLs with the CR Framework together with the Metric for ‘Incremental Capability’.

- Apply and test the CRLs with the CR Framework and the Metric against an industrial based case-study.

- Refine the CRLs, the CR Framework and the Metric, as necessary, to give an assessment and measurement method a wider applicability. This is directly linked to a reduction in Risk.

- Design, develop and test a systematic index for ‘System Maturity’; ‘System Readiness’; and ‘Capability Readiness’.

- Extend the ‘System Readiness’ Boolean scale (0 or 1) to include other relevant factors such as ‘robustness’ of use. This scale could be applied across the attributes or components of ‘System Readiness’ which first may need to be defined (this could be equivalent to the Defence Lines of Development (DLoD) – TEPIDOIL for example).

- “Readiness” is context specific, but what about for different and multiple contexts?

11 Acknowledgements

The authors would like to thank colleagues Mr. Tim Mackley, Senior Lecturer; Dr. Steve Barker, Research Officer; and Dr. John Deane, Research Officer from the Department of Systems Engineering and Human Factors, School of Engineering at Cranfield University (UK) for their advice, encouragement and for reviewing the draft version of this paper.

12 References

[01] Mankins, J. C. 1995, “Technology Readiness Levels”, White Paper, Advanced Concepts Office, Office of Space Access and Technology, NASA, available at: http://www.hq.nasa.gov/office/codeq/trl/trl.pdf (accessed 29th November 2008).

[02] “Technology Readiness Level (TRL)”, from Wikipedia, the free encyclopaedia, available at: http://en.wikipedia.org/wiki/Technology_Readiness_Level (accessed 29th November 2008).

[03] “System Readiness Levels (SRLs)”, AOF Technology Management Policy, Information and Guidance on the Technology Management aspects of UK MOD Defence Acquisition version 1.0.1 – September 2008, available at: http://www.aof.mod.uk/aofcontent/tactical/techman/content/srl_whatarethey.htm (accessed 29th November 2008).

[04] “System Readiness Levels (SRLs)”, (September 2008), AOF Technology Management Policy, Information and Guidance on the Technology Management aspects of UK MOD Defence Acquisition version 1.0.1, available at: http://www.aof.mod.uk/aofcontent/tactical/techman/content/srl_applying.htm (accessed 29th November 2008).

[05] Gove, R., B. Sauser, J. Ramirez-Marquez (2008), “Technology Integration Maturity Metrics: Development of an Integration Readiness Level”, *International Journal of Technology Management* (in review), available at: http://personal.stevens.edu/~bsauser/Publications_files/Gove%202008%20IJTM.pdf (accessed 29th November 2008).

[06] "Manufacturing Readiness Levels - A New Tool for Technology Development and Acquisition Program Manufacturing Risk Reduction", U.S. Air Force Fact Sheet, available at: http://www.wpafb.af.mil/library/factsheets/factsheet_print.asp?fsID=9759&page=1 (accessed 29th November 2008).

[07] "Manufacturing Readiness Assessments", US Defense Acquisition University, Acquisition Community Connection, Where the DoD AT & L Workforce Meets to Share Knowledge, available at: https://acc.dau.mil/CommunityBrowser.aspx?id=18231 and https://www.mrlassist.bmpcoe.org/mrlassist/ (accessed 29th November 2008).

[08] Kober, B. J. and Sauser, B. (2008), "A Case Study in implementing a System Maturity Metric", *29th American Society of Engineering Management Conference*, West Point, NY.

[09] Sauser, B. (2008), "System Maturity indices for Decision Support in the Defense Acquisition Process", *5th Annual Acquisition Research Symposium of the Naval Postgraduate School.*

[10] Sauser B. et al., (2008), "A System Maturity index for the Systems Engineering Life Cycle", *International Journal of Industrial and Systems Engineering*, Vol. 3, No. 6.

[11] Sauser B. et al., (April 7th – 8th 2006), "From TRL to SRL: The Concept of Systems Readiness Levels", *Conference on Systems Engineering Research, Los Angeles, CA*, available at: http://personal.stevens.edu/~bsauser/Publications_files/Sauser%202006%20CSER.pdf (accessed 29th November 2008).

[12] "Technology Readiness Levels (TRLs)", (September 2008), AOF Technology Management Policy, Information and Guidance on the Technology Management aspects of UK MOD Defence Acquisition version 1.0.1, available at: http://www.aof.mod.uk/aofcontent/tactical/techman/content/trl_whatarethey.htm (accessed 29th November 2008).

[13] Sauser, B. et al., (2006), "Determining System Interoperability using an Integration Readiness Level", Stevens Institute of Technology, Systems Engineering and Engineering Management, available at: http://www.dtic.mil/ndia/2006systems/Thursday/sauser.pdf (accessed 29th November 2008).

[14] "System Readiness Levels (SRLs)", (September 2008), AOF Technology Management Policy, Information and Guidance on the Technology Management aspects of UK MOD Defence Acquisition version 1.0.1, available at: http://www.aof.mod.uk/aofcontent/tactical/techman/content/srl_whatarethey.htm (accessed 29th November 2008).

[15] Dowling, T. and Pardoe T., (2005), *"TIMPA – Technology Insertion Metrics",* Volume 1, Ministry of Defence, United Kingdom, QinetiQ.

[16] Sauser, B. et al., (2008), "A Probabilistic Approach to Maturity Assessment", School of Systems and Enterprises, Stevens Institute of Technology, available at: http://personal.stevens.edu/~bsauser/Publications_files/TMC%202008%20Sauser%20et%20al.pdf (accessed 29th November 2008).

[17] "CMMISM for Systems Engineering/Software Engineering", (November 2008), Version 1.02 (CMMI-SE/SW, V1.02), Software Engineering Institute, Carnegie Mellon University.

[18] "Draft Systems Engineering Standard", (1994), AFMC, MIL-STD 499B, Systems Engineering, 5/6/92.

[19] "What is Through Life Capability Management?", (October 2008), AOF Acquisition Operating Framework, The high level principles and information that guide how the UK MOD Defence Acquisition Community work version 2.0.6, available at: http://www.aof.mod.uk/aofcontent/operational/business/capabilitymanagement/capabilitymanagement_whatis.htm (accessed 29th November 2008).

[20] Petee, D. A. and Componation, P. J., (September 2003), "Predicting USMC SBIR Phase 1 to II Transition Success by Evaluating Use of Systems Engineering Capabilities", *EMJ – Engineering Management Journal*, Vol. 15, No. 3.

[21] *"*Defence Lines of Development*",* (December 2008), AOF, Acquisition Operating Framework, The high level principles and information that guide how the UK MOD Defence Acquisition Community works version 2.0.8, available at: http://aof.mod.uk/aofcontent/strategic/guide/sg_dlod.htm (accessed 29th November 2008).

13 Biographies

Abideen Tetlay is a PhD Systems Engineering researcher in the School of Engineering at Cranfield University (UK). His doctorate is funded by the UK Engineering and Physical Sciences Research Council (EPSRC) and BAE Systems. He has spent 10 years working in the IT industry as a professional Software and Database Engineer; reaching Consultant level. He has worked for the following UK based companies: K3 Business Technology Group; Bidwells Property Consultants; Lloyds Banking Group; and the Royal Society of Chemistry. Before embarking on an IT career, he first obtained a Bachelor's degree BA (Hons) Business Administration from the University of Bedfordshire (UK). This was followed by a Master's degree MSc Information Technology (Management Information Systems) from Cranfield University (UK) and then he moved into the IT sector. Several years later, he gained another Master's degree MSc Software Engineering from St Cross College, University of Oxford (UK); which was sponsored by Lloyds Banking Group. He is a professional member of the British Computer Society (BCS) and the Institution of Analysts and Programmers (IAP).

Philip John joined Cranfield University (UK) in 1999 as the Professor of Systems Engineering and is the Head of the Department of Systems Engineering and Human Factors. Following his PhD at Imperial College, London he spent 18 years in industry, holding a wide range of systems engineering and management roles, including Head of Systems Engineering for a major multinational company. His experience and responsibilities in industry encompassed the whole scope of systems engineering, including Requirements Engineering, System Design, ILS, ARM, Human Factors, Safety, Systems Proving & Simulation and Modelling. He is a member of several National Advisory Committees and Industrial Steering Boards and served as the President of the International Council on Systems Engineering (INCOSE) in the UK from 2003 to 2004. His research interests include: Understanding Complex Systems and Systems of Systems (SoS); Managing Complex Systems Projects and Risks; Through Life Capability Management; and Coping with Uncertainty and Change in Systems.